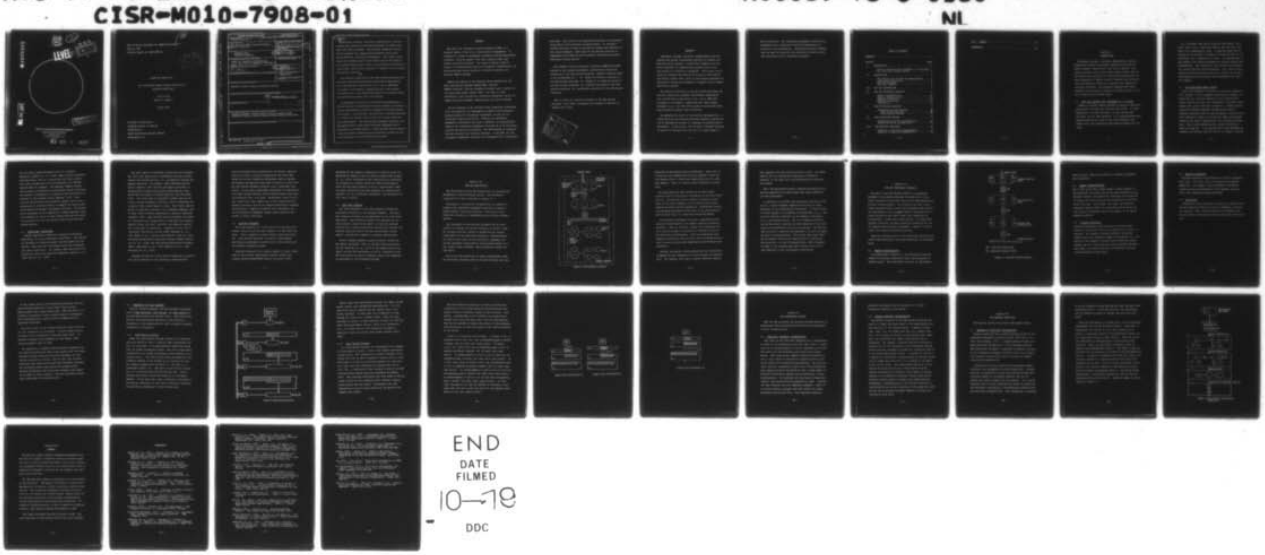
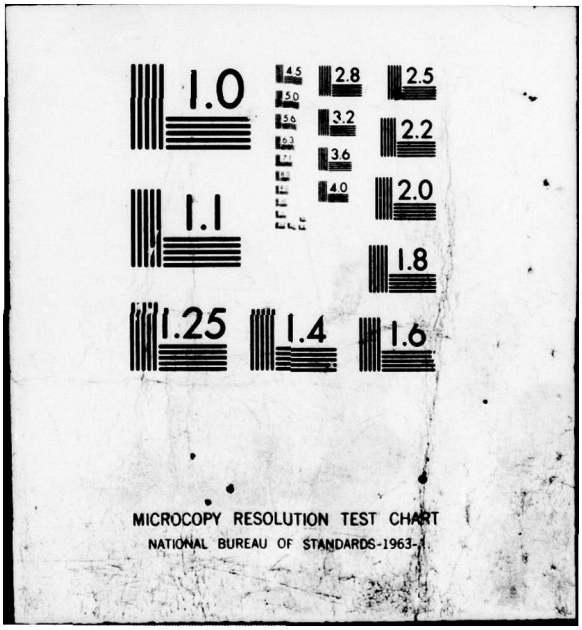
AD-A073 485 ALFRED P SLOAN SCHOOL OF MANAGEMENT CAMBRIDGE MA CEN--ETC F/G 9/2
THE INTELLIGENT MEMORY SYSTEM ARCHITECTURE - RESEARCH DIRECTION--ETC(U)
AUG 79 C LAM, S E MADNICK N00039-78-G-0160
UNCLASSIFIED CISR-M010-7908-01 NL

1 OF 1
AD A073485



END
DATE FILMED
10-79
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

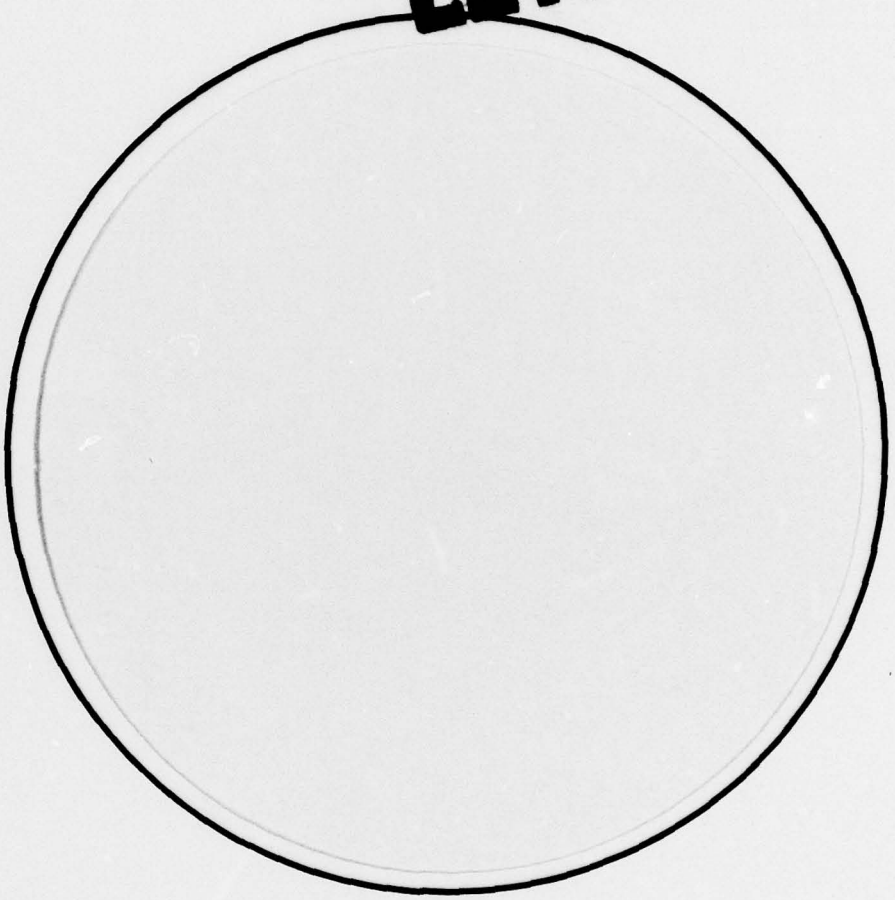
12

DDC
RECEIVED
SEP 7 1979
C

LEVEL II

AD A073485

DDC FILE COPY



DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

Center for Information Systems Research

Massachusetts Institute of Technology
Alfred P. Sloan School of Management
50 Memorial Drive
Cambridge, Massachusetts, 02139
617 253-1000

79 09 6 027

(12)

Basic Ordering Agreement No. N00039-78-G-0160

Task No. 003

Internal Report No. M010-7908-01

Technical Report No. 1

The Intelligent Memory System Architecture - Research Directions

Chat-Yu Lam

Stuart E. Madnick

August 1979

Principal Investigator:

Professor Stuart E. Madnick

Prepared for:

Naval Electronics Systems Command

Washington, D.C.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: Technical Report No. 1	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): (16) The Intelligent Memory System Architecture - Research Directions		5. TYPE OF REPORT & PERIOD COVERED: (9) Technical repts
		6. PERFORMING ORG. REPORT NUMBER: MIT-7908-01
7. AUTHOR(s): (10) Chat-Yu Lam, Stuart E. Madnick		8. CONTRACT OR GRANT NUMBER(s): (15) N00039-78-G-0160-C01
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Center for Information Systems Research, M.I.T. Sloan School of Management, Cambridge, MA 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: (12) 46 p.		(11) 12. REPORT DATE: August 1979 13. NUMBER OF PAGES: 43
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report): UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

(14) CISR-MIT-7908-01, CISR-IR-1

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

database management, database computer, multiple processor system, hierarchical system, C^3 system, storage hierarchy, automatic data repair

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

409 590

20. Abstract

Information storage, retrieval, communication, and processing have become increasingly important for modern command and control systems. Conventional computers are primarily designed for computational purposes and are not well-suited for information management. This report introduces the concepts and research directions of the Intelligent Memory System (IMS), which is particularly designed for large-scale information management to support future command and control systems.

The specific objectives of the IMS include providing substantial information management performance improvements over conventional architectures (e.g., up to 1000-fold increases in throughput), supporting very large complex databases (e.g., over 100 billion bytes of structured data), and providing extremely high reliability.

By applying the theory of hierarchical decomposition, a highly parallel and pipelined database computer architecture can be implemented by means of complexes of multiple-microprocessors. In particular, IMS utilizes a storage hierarchy to handle the storage and retrieval of a large volume of data efficiently. The information management functions are decomposed into a functional hierarchy implemented by a hierarchy of microprocessors. Decentralized control mechanisms are used to coordinate the activities of both the storage hierarchy and the functional hierarchy.

Preface

The Center for Information Systems Research (CISR) is a research center of the M.I.T. Sloan School of Management. It consists of a group of management information systems specialists including : faculty members, full-time research staff, and student research assistants. The Center's general research thrust is to devise better means for designing, implementing, and maintaining application software, information systems, and decision support systems.

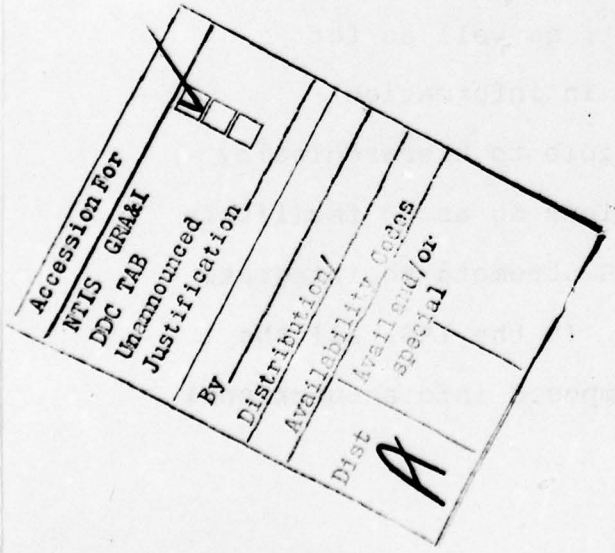
Within the context of the research effort sponsored by the Naval Electronics Systems Command under contract N00039-78-G-0160, CISR has proposed to conduct basic research on the Intelligent Memory System (IMS). The IMS is a high performance, high availability information management system for supporting future Command, Communication and Control Systems.

Current advances in LSI and Multi-Chip Integration technology offer the potential for development of modular multi-processor building blocks for information management, as well as for intelligent memory controllers. Advances in information management technologies have made it possible to hierarchically organize the information management functions so as to facilitate pipeline and parallel processing. The IMS attempts to integrate the above hardware and software advances. In the IMS, all the information management functions are decomposed into a functional

hierarchy. Each level of the functional hierarchy is implemented using modular multi-processor building blocks. An automatic storage hierarchy is used by the IMS for storage and retrieval of very large databases. Each level of the storage hierarchy is implemented using modular multi-processor controllers and their associated storage devices.

The proposed research described in Contract N00039-78-G-0160 focuses on the concept development, architectural design and evaluation of the IMS storage hierarchy. Specific research tasks to be accomplished are : (1) design of a general structure of the IMS storage hierarchy, (2) design of a revised structure of the IMS storage hierarchy, (3) develop algorithms for the IMS storage hierarchy, (4) performance evaluation of the IMS storage hierarchy.

This is first of a series of reports on the IMS storage hierarchy. This report introduces the concepts of IMS and its research directions.



ABSTRACT

Information storage, retrieval, communication, and processing have become increasingly important for modern command and control systems. Conventional computers are primarily designed for computational purposes and are not well-suited for information management. This report introduces the concepts and research directions of the Intelligent Memory System (IMS), which is particularly designed for large-scale information management to support future command and control systems.

The specific objectives of the IMS include providing substantial information management performance improvements over conventional architectures (e.g., up to 1000-fold increases in throughput), supporting very large complex databases (e.g., over 100 billion bytes of structured data), and providing extremely high reliability.

By applying the theory of hierarchical decomposition, a highly parallel and pipelined database computer architecture can be implemented by means of complexes of multiple-microprocessors. In particular, IMS utilizes a storage hierarchy to handle the storage and retrieval of a large volume of

data efficiently. The information management functions are decomposed into a functional hierarchy implemented by a hierarchy of microprocessors. Decentralized control mechanisms are used to coordinate the activities of both the storage hierarchy and the functional hierarchy.

TABLE OF CONTENTS

Section I

INTRODUCTION

Information storage, retrieval, communication, and processing have become increasingly important activities of modern command, communication, and control systems. The primary objective of the Intelligent Memory System (IMS) is the design and evaluation of a specialized architecture for data management to support future C3 systems that will provide functionality, speed and reliability far beyond that currently available. An integrated approach that takes advantage of the current hardware and software state of the art is used to attain these goals.

1.1 NEED FOR ADVANCED DATA MANAGEMENT IN C3 SYSTEMS

Modern C3 systems require substantial amount of data processing which includes the following activities: (1) data acquisition, (2) data reduction, (3) data storage and retrieval, and (4) data analysis. As C3 systems become more sophisticated and are used to meet more demanding challenges, their data processing requirements become more critical.

In a C3 system, data may be acquired from sensors (e.g., radar, sonar, etc.), from human inputs, and from other computers over communication links. A reduction phase may be carried out on the data to reduce its volume and improve its usefulness (e.g. elimination of noise). The reduced data is then added to the various databases for further analysis and dissemination. Data analysis is the retrieval and processing of data from the various databases. The result of data analysis provides feed back controls to sensors for further data acquisition and provides timely information for decision support.

1.2 THE INTELLIGENT MEMORY SYSTEM

Since data management functions are essential components of C3 systems, a data management system must be capable of supporting the increasing demand for more functionality, larger capacity for data, more responsive in handling larger system load, and provide more reliability. Though current commercial database mangement systems provide fairly advanced functionalities for data management, they are built upon traditional computer architectures which are designed primarily for numeric computations, not information management. Thus, these systems provide limited reliability, speed and capacity. A new approach that takes advantage of hardware and software state of the art is needed to realize

a viable data management system to support future C3 systems.

Madnick (Madnick, 1973) proposed a new approach to structure a data management system. This approach makes use of the techniques of hierarchical decomposition to modularize all the functions within a data management system so that the entire system can be implemented using a microprocessor complex. This approach has been referred to as INFOPLEX in the literature. Research on INFOPLEX has been focused on developing the theory and methodologies for decomposing the data management functions into a hierarchy of functions (Huff and Madnick, 1979), and on the analysis of theoretic properties of data movement algorithms in a generalized storage hierarchy (Lam and Madnick, 1979).

The IMS architecture is based upon the INFOPLEX concepts and theories. The research focus of IMS is directed at studying the architecture of a specific data management hardware/software system which makes use of multiple-processor modules as its basic building block. The IMS is a critical step toward a data management systems capable of effectively supporting future C3 systems.

Section II

RELATED WORK

In the past, computers were designed primarily for computational purposes. In recent years, several ideas have been suggested for modifying the computer system architecture in order to handle information processing more efficiently. These ideas can be largely divided into the following four approaches: (1) new instructions through microprogramming, (2) intelligent controllers, (3) dedicated minicomputers for database operations, (4) specialized database computers. Most of the previous research activities in the field have been focused on the first three categories. The IMS architecture belongs to the fourth category but also incorporates features from all four approaches. In the following, these four approaches are briefly described.

2.1 NEW INSTRUCTIONS THROUGH MICROPROGRAMMING

Conventional computer instructions are usually not well suited to the requirements of database systems. Using firmware, it is possible to augment or enhance the instructions. This approach has been adopted in several systems. One of the earliest efforts occurred as part of the LISTAR informa-

tion retrieval system developed at M.I.T.'s Lincoln Laboratory (Armenti et. al., 1970), where several frequently used operations, such as a generalized List Search operation, were incorporated into the microcode of an IBM System/360 Model 67 computer. The Honeywell H60/64 incorporates special instructions to perform data format conversion and hashing corresponding to frequently used subroutines of Honeywell's IDS database system (Bachman, 1975). More recently IBM announced the System/38 computer where microprogramming is used extensively to support the various operating system and database management functions. The performance advantages of this approach are highly dependent upon the frequency of use of the new instructions and the extent to which they fit into the design of the overall database system software.

2.2 INTELLIGENT CONTROLLERS

Another approach to improving information processing efficiently is to use intelligent controllers. The controller provides an interface between the main memory and the devices. More and more intelligence have been introduced into these controllers. For example, many controllers can perform the search key operation themselves (Ahern et. al., 1972; Lang et. al., 1977).

Two major types of intelligent controllers have emerged. The first type specializes in automating the data transfer between the storage devices, i.e., the physical storage management functions. For example, IBM's 3850 Mass Storage System (Johnson, 1975) uses an intelligent controller to automatically transfer data between high-capacity, slow-speed tape cartridges and medium-capacity, fast move-head disks. The second type is designed to handle some of the logical storage management functions, such as searching for a specific data record based on a key. This latter type of device is sometimes referred to as a database computer, and is often used to perform associative or parallel searching (Langdon, 1978). Most parallel search strategies are based on a head-per-track storage device technology (for example, magnetic drums, LSI shift registers, and magnetic bubbles) and a multitude of comparators. Examples of this type of intelligent controllers include CASSM (Copeland et. al., 1973; Healy et. al., 1972; Su and Lipovski, 1975; Su, 1977), the Rotating Associative Relational Storage (RARES) design (Lin et. al., 1976), and the Rotating Associative Processor (RAP) (Ozkarahan et. al., 1975; Schuster et. al., 1976; Ozkarahan et. al., 1977).

Although the decline in the costs of comparator electronics, due to advances in LSI technology, makes parallel

search strategies quite promising for the future, they are only well suited to storage technologies that lend themselves to low cost read/write mechanisms, and for optimal performance and operation they tend to require a fairly simple and uniform database structure (e.g., relational flat files). To use these intelligent controllers in conjunction with other storage devices, such as mass storage, some staging mechanisms have to be used. Furthermore, these intelligent controllers only support part of the information management functions, much of the complex functions of language interpretation, support of multiple user interfaces, etc., of an information management system cannot easily be performed in these controllers.

2.3 BACK-END PROCESSOR

The third approach is to shift the entire database management function from the main computer to a dedicated computer. Such a computer is often called a back-end processor. The back-end processor is usually a minicomputer specifically programmed to perform all of the functions of the database management system.

Back-end processors have evolved rapidly in recent years. Some of the earliest experimental efforts include the loosely coupled DATACOMPUTER (Marill and Stern, 1975),

developed by the Computer Corporation of America using the DECSystem-10 computer, and the tightly coupled XDMS (Canady et. al., 1974), developed by Bell Laboratories by modifying the firmware of a Digital Scientific META-4 minicomputer. Since the back-end processor is still a conventional computer whose architecture has been designed for computational purposes, not for information management, its performance is still quite limited.

2.4 DATA BASE COMPUTER

The fourth approach to providing improved information processing efficiency is the database computer. The database computer has most of the advantages of the first three approaches. The difference between this approach and the third approach (back-end processor) is that the database computer has a system architecture particularly suitable for database operations while the back-end processor merely adapts a conventional computer to database applications.

Current database computer research efforts include the DBC (Hsiao and Kannan, 1976) at the Ohio State University, the GDS (Hakozaki et. al., 1977) at the Nippon Electric Co., Japan, and the IMS architecture at M.I.T. Features of the IMS architecutre as well as research issues to be addressed are discussed in the following sections.

Section III

THE IMS ARCHITECTURE

The key concepts of the IMS architecture are hierarchical decomposition and distributed control. The conceptual organization of IMS is depicted in Figure 3.1.

Techniques of hierarchical decomposition are applied to organize the information management functions to obtain a highly modular functional hierarchy. Each level of the functional hierarchy is implemented using multiple microprocessors.

The techniques of hierarchical decomposition are also applied to organize the storage subsystem to obtain a modular storage hierarchy capable of supporting the storage requirements of the functional hierarchy. Microprocessors are used at each level of the hierarchy to implement the storage management algorithms so the hierarchy appears as a very large, highly reliable, high performance virtual storage space.

Due to the high modularity of these organizations, both the functional hierarchy and the storage hierarchy can take

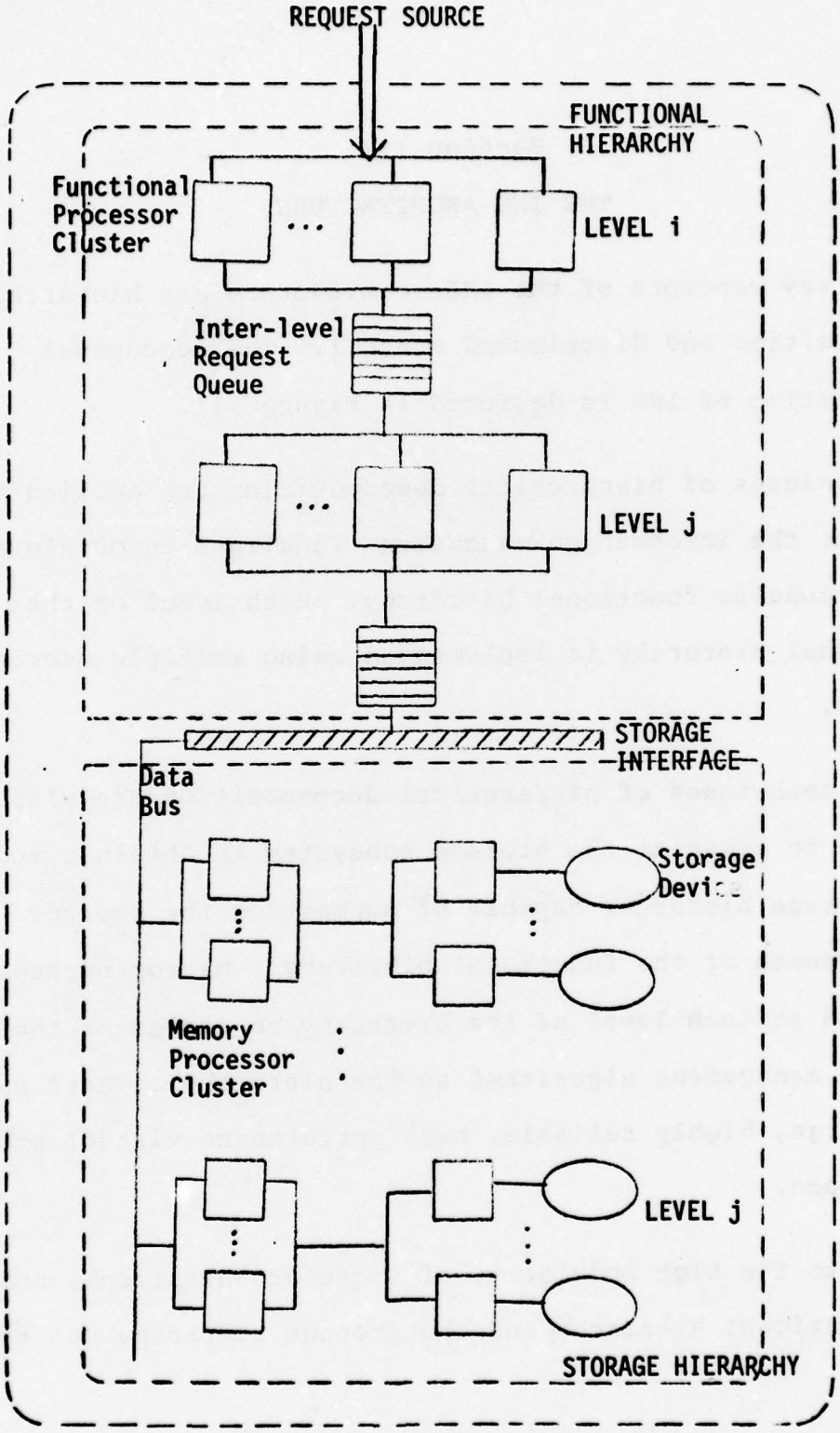


Figure 3.1 IMS Conceptual Structure

advantage of distributed control mechanisms. Each level in a hierarchy only communicates with its adjacent levels and each module within a level only communicates with its adjacent modules. Thus, no central control mechanism is necessary.

The high speed and high reliability of IMS follows directly from the hierarchical decomposition and distributed control. Distributed control enhances reliability since there is no single component in the system whose failure renders the entire system inoperative. Distributed control also enhances performance since there is no system bottleneck as would exist in a centrally controlled system.

A functionally decomposed hierarchy, implemented using multiple microprocessors, can support pipeline processing naturally. That is, multiple requests for information can be at various stages of processing at different levels of the hierarchy simultaneously. Such an architecture also enhances reliability since an error can be isolated within a level in the hierarchy thus simplifying error detection and correction.

Parallel processing is made possible by the hierarchical decomposition and implementation using multiple microprocessors. For example, there may be several identical modules

that implement the same function within a level. All these modules can be simultaneously operating on different requests, at the same time, providing potential backup for one another.

Thus, the distributed control, pipeline and parallel processing capabilities of IMS provide very high reliability and high performance.

In addition to providing high performance and high reliability, a viable database computer must be able to take advantage of new technological innovations. It must be able to easily upgrade to incorporate new algorithms, e.g., a new security checking technique, or new hardware innovations, e.g., a new storage device. Due to its modular structure, the IMS functional hierarchy can take advantage of new techniques and technologies as they are developed. The IMS storage hierarchy is specifically designed to be able to handle any type of storage devices. Thus unlike some other information systems, which may be specialized to a particular data structure, or type of storage device, IMS is designed to adapt to the changing application needs as well as to take advantage of new technological innovations.

Section IV

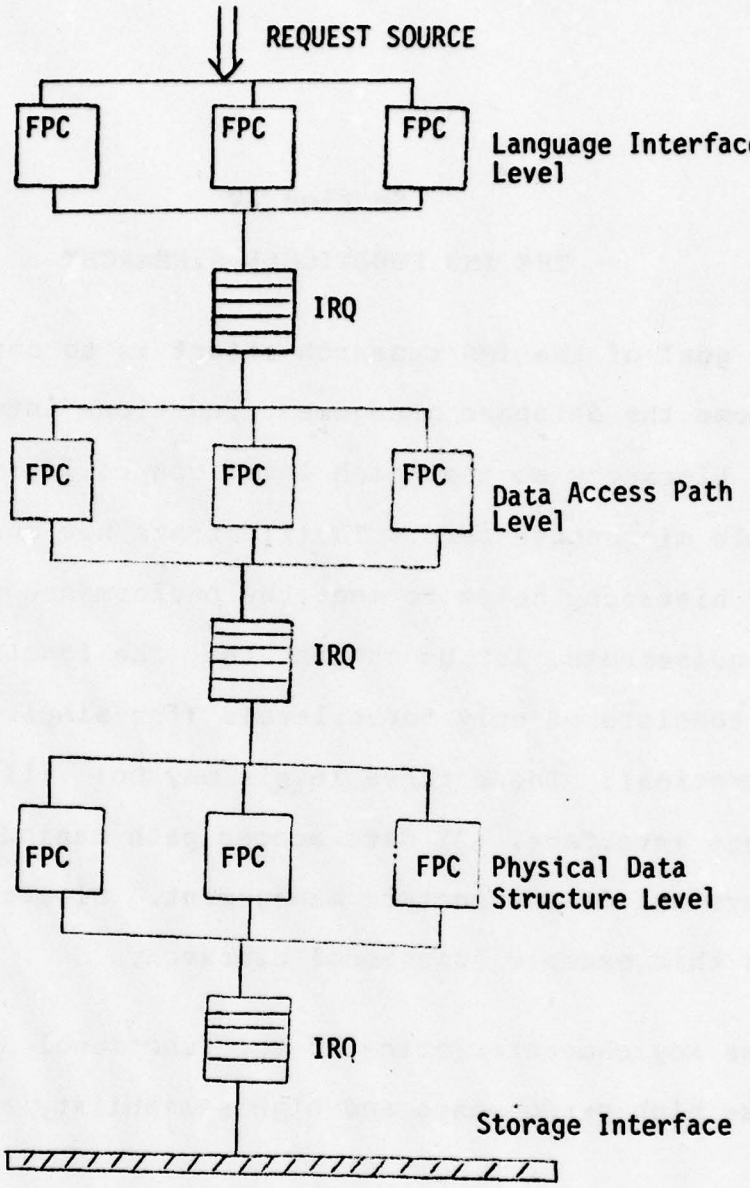
THE IMS FUNCTIONAL HIERARCHY

One goal of the IMS research effort is to optimally decompose the database management functions into a functional hierarchy so that each level can be implemented using multiple microprocessors. To illustrate how such a functional hierarchy helps to meet the performance and reliability requirements, let us suppose that the functional hierarchy consists of only three levels (for simplicity of illustration). These three levels may be: (1) high-level language interface, (2) data access path manipulation, and (3) physical data structure management. Figure 4.1 illustrates this example functional hierarchy.

Some key characteristics of the functional hierarchy that realize high performance and high reliability are discussed below.

4.1 MODULAR IMPLEMENTATION

As illustrated in Figure 4.1, the functional hierarchy consists of multiple functional levels interconnected by request queues. Each functional level can be implemented



IRQ : Inter-level Request Queue

FPC : Functional Processor Cluster

Figure 4.1 Functional Hierarchy Example

using multiple identical processors to enhance performance and reliability.

4.2 DYNAMIC RECONFIGURATION

Due to the use of only a small number of basic modules in the functional hierarchy, and the way these modules are constructed, dynamic reconfiguration techniques can be applied to enhance the reliability and the performance of the functional hierarchy. For example, the number of Functional Processor Clusters (FPCs) in a functional level can be varied to meet the changing load of the system, or to replace defective modules.

4.3 PIPELINE PROCESSING

Pipeline processing is facilitated by the hierarchical structure, since each level can be regarded as a stage in a pipeline. For example, multiple different information requests may be at different stages of processing, e.g., one may be at the language interface being checked for syntax, another may be at the access path level searching through a hash table, yet another may be at the physical structure level allocating storage space.

4.4 PARALLEL PROCESSING

Multiple identical microprocessors are used to implement each level. This gives rise to the parallel processing capability. For example, several different information requests may be simultaneously in the language interface level, each undergoing syntax processing.

4.5 RELIABILITY

Reliability of the functional hierarchy is enhanced since multiple identical modules are used to implement a particular function. Thus, a failed module can be removed from service without interrupting continous operation of the system.

Section V

THE IMS STORAGE HIERARCHY

To provide a high performance, highly reliable, and large capacity storage subsystem, IMS makes use of an automatically managed storage hierarchy.

The effectiveness of a storage hierarchy depends heavily on the phenomonon known as locality of reference (Denning, 1970). A storage hierarchy makes use of this property of information reference pattern so that the information that is most likely to be referenced is found in the higher levels of the hierarchy, giving the storage hierarchy an expected access time close to the expected access time of the faster memories. This approach has been used in contemporary computer systems in cache memory systems, in virtual memory demand paging systems, and in mass storage systems.

Figure 5.1(a) illustrates the general structure of an example storage hierarchy consisting of six levels. The key characteristics of some of the devices that can be used in these levels are presented in Figure 5.1(b).

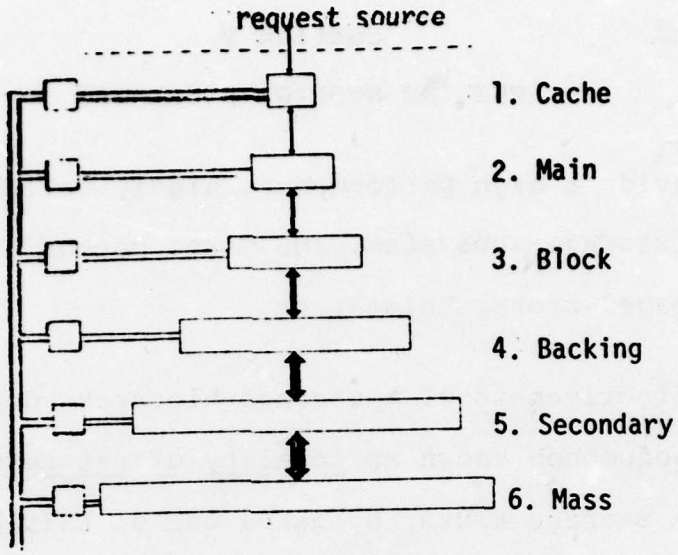


Figure 5.1(a) Memory Hierarchy Example

Storage Level	Random Access Time	Sequential Transfer Rate (bytes/sec)	Unit Capacity (bytes)	System Price (per byte)
1. Cache	100 ns	100M	32K	50c
2. Main	1 μs	16M	512K	10c
3. Block	100 μs	8M	2M	2c
4. Backing	2 ms	2M	10M	0.5c
5. Secondary	25 ms	1M	100M	0.02c
6. Mass	1 sec.	1M	100B	0.0005c

Figure 5.1(b) Range of Storage Devices

To the lowest level of the functional hierarchy, the storage hierarchy appears as a very large linear virtual address space with a small access time. The fact that it is actually a hierarchy or that a certain block of information can be obtained from a certain level is hidden from the functional hierarchy.

The lowest level of the storage hierarchy always contains all the information of the system. A level always contains a subset of the information in the next lower level. To satisfy a request, the information in the highest (most easily accessed) level is used.

The capability to support many functional processors and the inherent parallelism within each storage level and among different storage levels provide high throughput for the storage hierarchy as a whole. Use of novel data movement algorithms that maintain multiple data redundancy and makes use of multiple block sizes across the storage levels enhances performance and facilitates automatic data repair in the event of device failure. The key strategies used by these algorithms are discussed below.

5.1 STRATEGIES FOR DATA MOVEMENT

Various storage management and data movement techniques, such as page splitting, read through, and store behind can be distributed within the storage levels. This facilitates parallel and asynchronous operation in the hierarchy. Furthermore, these approaches can lead to greatly increased reliability of operation.

5.1.1 Read Through Strategy

Under the read-through strategy (Figure 5.2), when data currently stored at level i (and all lower performance levels) is referenced, it is simultaneously copied and stored into all higher performance levels. The size of data being broadcasted to all higher performance levels depends on the storage level i. The size of data being extracted from the broadcast depends on the storage level receiving the data. For example, suppose that the datum 'a', at level 3, is referenced (Figure 5.2). The block of size N(2,3) containing 'a' is extracted and moved up the data bus. Level 2 extracts this block of data and stores it in its memory modules. At the same time, level 1 extracts a sub-block of size N(1,2) containing 'a' and level 0 extracts a sub-block of size N(0,1) containing 'a' from the data bus.

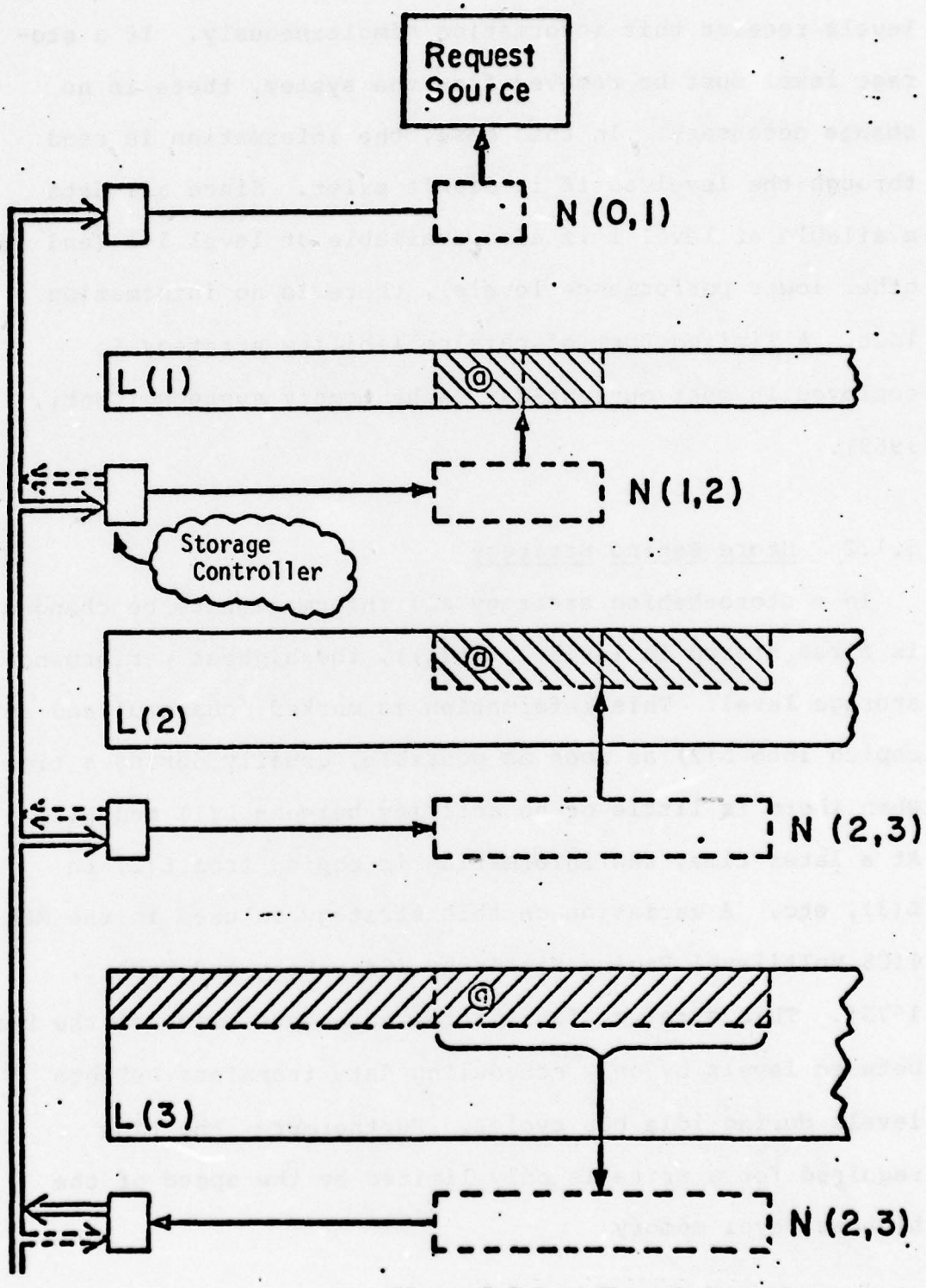


Figure 5.2 Read-Through Operation

Hence, under the read-through strategy, all upper storage levels receive this information simultaneously. If a storage level must be removed from the system, there is no change necessary. In this case, the information is read through the level as if it didn't exist. Since all data available at level i is also available at level i+1 (and all other lower performance levels), there is no information lost. A limited form of this reliability strategy is employed in most current-day cache memory systems (Conti, 1969).

5.1.2 Store Behind Strategy

In a store-behind strategy all information to be changed is first stored in level 1, (L(1)), the highest performance storage level. This information is marked 'changed' and is copied into L(2) as soon as possible, usually during a time when there is little or no activity between L(1) and L(2). At a later time, the information is copied from L(2) to L(3), etc. A variation on this strategy is used in the MULTICS Multilevel Paging Hierarchy (Greenberg and Webber, 1975). This strategy facilitates more even usage of the bus between levels by only scheduling data transfers between levels during idle bus cycles. Furthermore, the time required for a write is only limited by the speed of the highest level memory.

The store-behind strategy can be used to provide high reliability in the storage system by maintaining multiple copies of data in different levels of the hierarchy. Ordinarily, a changed page is not allowed to be purged from a storage level until the next lower level has been updated. This can be extended to require two levels of acknowledgment thus maintaining at least two copies of the same information in the system.

Figure 5.3 illustrates this process. In Figure 5.3(a), a processor stores into L(1), the corresponding page is marked 'changed' and 'no lower level copy exists'. In Figure 5.3(b), at a later time, the corresponding page in L(2) is updated and marked 'changed' and 'no lower level copy exists'. An acknowledgment is sent to L(1) so that the corresponding page is marked 'one lower level copy exists'. At a later time, shown in Figure 5.3(c), the corresponding page in L(3) is updated and marked 'changed' and 'no lower level copy exists'. An acknowledgment is sent to L(2) so that the corresponding page is marked 'one lower level copy exists'. An acknowledgment is sent to L(1) so that the corresponfing page is marked 'two lower level copies exists'. At this time, the page in L(1) may be replaced if necessary, since then there will be at least two copies of the updated information in the lower memory levels.

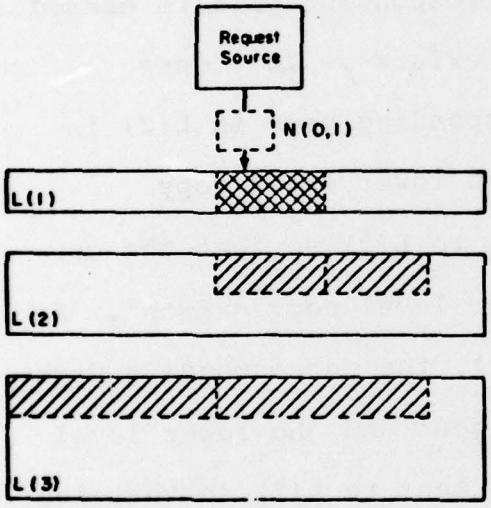


Figure 5.3(a) Store-behind (a)

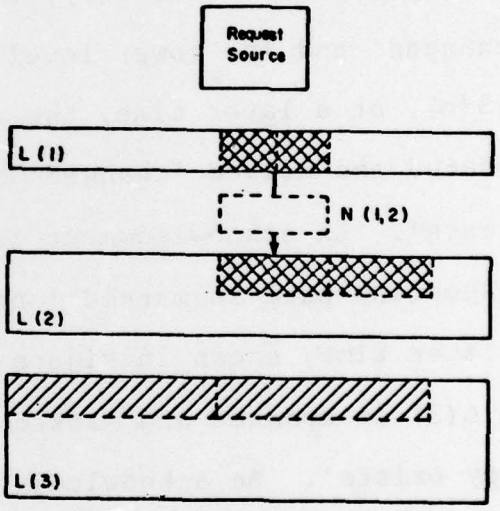


Figure 5.3(b) Store-behind (b)

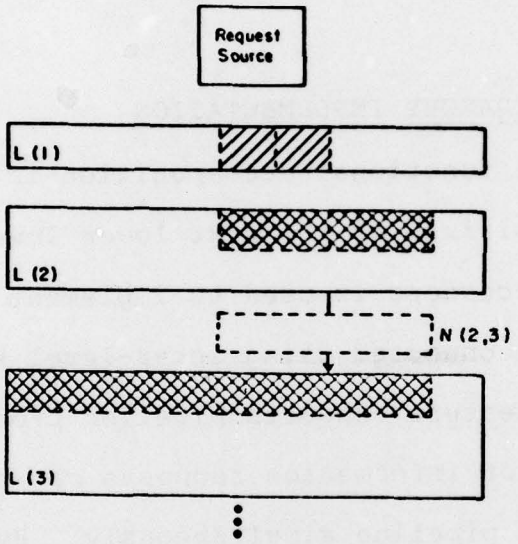


Figure 5.3(c) Store-behind (c)

Section VI

IMS DISTRIBUTED CONTROL

Both the IMS functional and storage hierarchy make use of distributed control mechanisms and multiple microprocessors in their implementations.

6.1 FUNCTIONAL HIERARCHY IMPLEMENTATION

Each level of the functional decomposition is implemented in terms of the primitives of the next lower level. A separate set of microprocessors is used to implement each level. The levels are interconnected using inter-level request queues. This architecture supports pipeline processing so that a large number of information requests can be at different stages of the pipeline simultaneously. By using multiple identical processors at each level, this architecture also supports parallel processing so that a large number of different information requests can be serviced at a level simultaneously. To coordinate the activities of these processors, distributed control mechanisms are used. A processor only interacts with its immediate neighbors. There is no central control processor that must control all other processors and the data flow. This approach eliminates

potential bottlenecks and the existence of critical reliability elements in the system.

6.2 STORAGE HIERARCHY IMPLEMENTATION

The general structure of the IMS storage hierarchy consists of a common data path which is the communication link that connects all the storage levels. A typical storage level 'taps' into the common data path via a common interface. The storage controllers at each storage level implements the algorithms that support distributed control mechanisms. For example, a storage controller maintains a directory of all the information in its associated storage devices and only responds to request for information that is in its associated storage devices. A request for sevice or a request for information is passed among the storage levels via the common data path until it is handled by the appropriate storage level. This asynchronous operation of the storage heirarchy and the use of distributed controls enhance performance by allowing a large number of storage requests to be handled at different storage levels and at different devices within a storage level simultaneously. Furthermore, reconfiguaration of the storage hierarchy is facilitated by the uniform interface to the common data path and the use of multiple processor modules as storage controllers at each level.

Section VII

IMS RESEARCH DIRECTIONS

This section outlines the overall IMS research focus.

7.1 RESEARCH IN FUNCTIONAL DECOMPOSITION

Various researchers have proposed decompositions for the information management functions (Senko, 1976; Toh et. al., 1977; Yeh and Baker, 1977). A common weakness of most proposed functional decomposition is that although they may make good sense and impose a reasonable conceptual structure on the information management function, there are no generally accepted criteria with which to evaluate or compare alternative decompositions.

A qualitative criteria often used to decompose complex functions into sub-modules is that of modularity. A decomposition is considered to attain high modularity when each individual module is internally coherent, and all the modules are loosely coupled with one another. In a related research effort we have formalized the notion of modularity and developed a methodology to systematically identify optimal functional decompositions. This methodology is referred

to as the Systematic Design Methodology (SDM) approach (Huff and Madnick,1978). In the IMS research, this methodology will be adapted to obtain an optimal IMS functional hierarchy.

Using the SDM approach, we begin with a set of functional requirements for the IMS functions (Step 1). Each pair of requirements is examined in turn, and a decision as to whether a significant degree of interdependency between the two requirements exists is made (Step 2). Then the resulting information is represented as a non-directed graph structure: nodes are requirement statements, links are assessed interdependencies (Step 3). The graph is then partitioned with the objective of locating a good decomposition (Step 4). A measure of goodness that takes into account the internal strength of a partition and the inter-partition strengths may be used. Once the optimal partioning is determined, the resulting subsets of requirements are interpreted as design sub-problems (Step 5). From these design sub-problems, a functional hierarchy for IMS can then be systematically derived (Step 6). These six steps are illustrated in Figure 7.1.

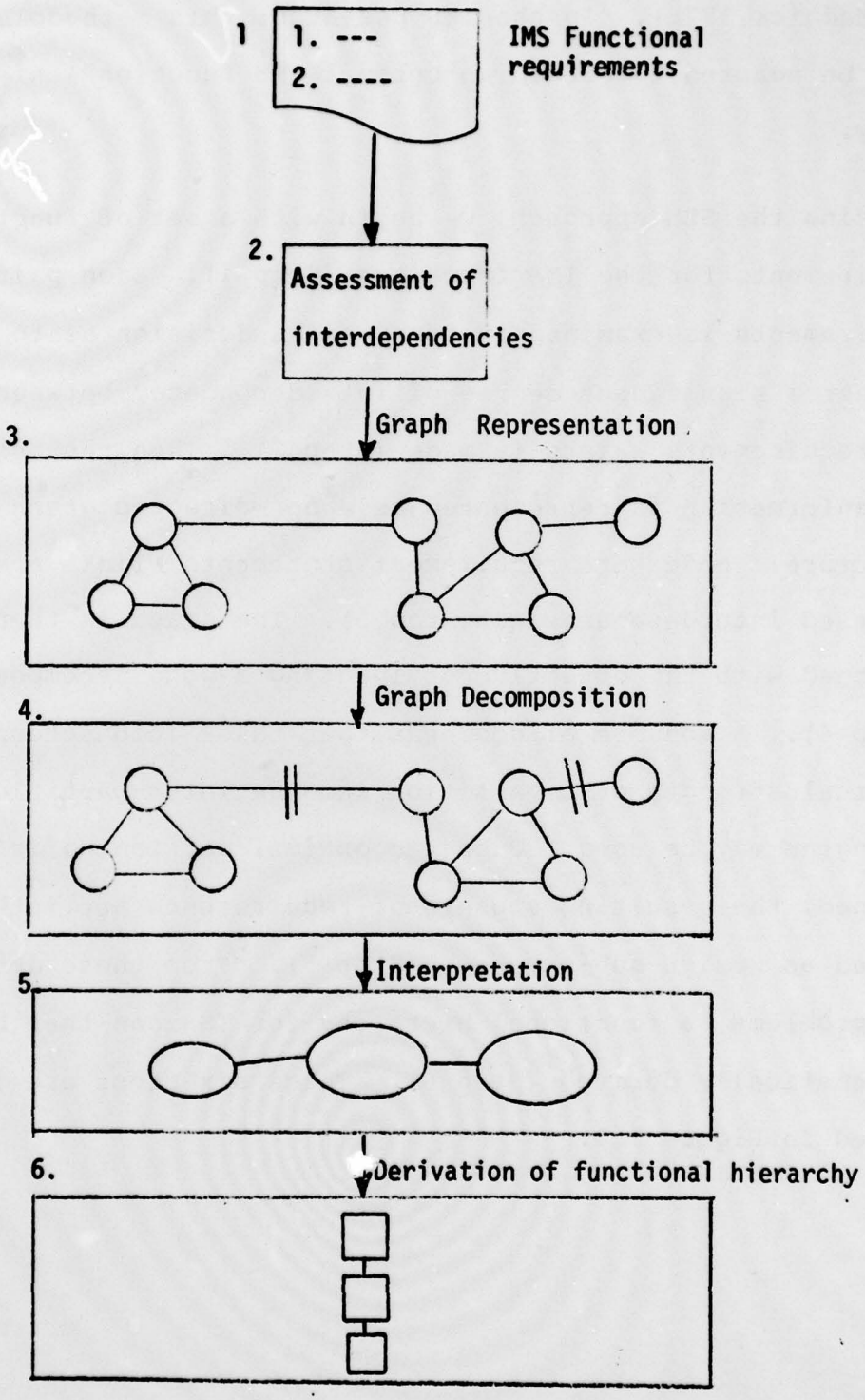


Figure 7.1 Use of SDM for IMS Functional Decomposition

7.2 RESEARCH IN PHYSICAL DECOMPOSITION

In the earlier INFOPLEX study we have formalized the data movement algorithms and developed formal models for a generalized storage hierarchy. Theoretic properties concerning these algorithms and the generalized storage hierarchy have been analyzed in detail (Lam and Madnick, 1979). The IMS physical decomposition research efforts will focus on the issues concerning the architecture of a practical storage hierarchy that incorporates some of these algorithms and theoretic results. A key concept in the IMS storage hierarchy design is multiple data redundancy. By using appropriate algorithms for data movement among levels of the storage hierarchy and with appropriate design for the storage hierarchy, multiple copies of the same data can be maintained across the storage levels. Techniques for automatic repair can be developed so that the IMS storage hierarchy is not susceptible to any single and most multiple device failures.

Two types of automatic repair algorithms are of particular interest. For certain critical data, multiple copies of it may be maintained within a single storage level. When a storage module fails, data on it may be reconstructed quickly from its copies in some other storage modules within the same storage level. This type of automatic repair

algorithm is referred to as intra-level automatic data repair algorithm. Another type of automatic data repair algorithms makes use of the fact that multiple copies of the same data is maintained across the storage levels. Data on a failed module at a level may be reconstructed from its copies from a different level. This type of algorithm is referred to as inter-level automatic data repair algorithm.

The research focus on the IMS physical decomposition can be summarized as the following tasks: (1) develop effective storage hierarchies that make use of multiple microprocessor modules as basic buliding block for storage controllers, (2) develop practical algorithms for data movement among the storage levels that incorporate the read-through and store-behind strategies, (3) develop automatic data repair algorithms that take advantage of the multiple data redundancy properties of the storage hierarchy, and (4) evaluate the algorithms and the architectures.

Section VIII

SUMMARY

The need for highly reliable information management systems that can support transaction volume and storage capacity several orders of magnitude higher than current information management systems calls for new architectures, both in information management functions and the hardware that supports these functions.

The IMS Data Base Computer architecture is a major effort in this direction. IMS applies the theory of hierarchical decomposition to realize a highly structured, modular architecture. The information management functions are structured as a hierarchy and a highly modular implementation of the functional hierarchy using multiple microprocessors entails high degrees of parallelism and reliability. An automatic storage hierarchy is used to support the high performance, high capacity storage requirements of IMS.

This report discusses the basic concepts of IMS. The main objectives of IMS research efforts are also outlined.

REFERENCES

(Ahearn et. al., 1972) : Ahearn, G.P., Dishon, Y., and Snively, R.N., 'Design Innovations of the IBM 3830 and 2835 Storage Control Units', IBMJRD, Vol. 16, No. 1 (January, 1972), 11-18.

(Armenti et. al., 1970) : Armenti, A., Galley, S., Goldberg, R., Nolan, J., and Scholl, A. 'LISTAR - Lincoln Information Storage and Associatve Retrieval System'. AFIP Conference Proceedings, 36, (1970), 313-322.

(Bachman, 1975) : Bachman, C. 'Trends in Database Management - 1975'. AFIP Conference Proceedings, 44, (1975), 569-576.

(Canaday et. al., 1974) : Canaday, R.H., Harrison, R.D., Ivie, E.L., Ryder, J.L., and Wehr, L.A. 'A Back-end Computer for Database Management'. CACM, 17,10 (October 1974), 575-584.

(Conti, 1969) : Conti, C.J. 'Concepts for Buffer Storage'. IEEE Computer Group News, March 1969, 6-13.

(Copeland et. al., 1973) : Copeland, G.P., Lipovski, G.J., and Su, S.Y.W. 'The Architecture of CASSM: A Cellular System for Non-numeric Processing'. Proceedings of First Annula Symposium on Computer Architecture, December, 1973, 121-128.

(Denning, 1970) : Denning, P.J. 'Virtual Memory'. ACM Computing Surveys, 2, 3 (September 1970), 153-190.

(Greenberg and Webber, 1975) : Greenberg, B.S., and Webber, S.H. 'MULTICS Multilevel Paging Hierarchy'. IEEE INTERCON, 1975.

(Hakozaki et. al., 1977) : Hakozaki, K., Makino, T., Mizuma, M., Umemura, M., and Hiyoshi, S., 'A Conceptual Design of a Generalized Database Subsystem', VLDB, 1977, 246-253.

(Healy et. al., 1972) : Healy, L.D., Doty, K.L., and Lipovski, G.J. 'The Architecture of a Context Addressed Segment Sequential Storage'. AFIPS Conference Proceedings, 41, (1972), 691-701.

(Hsiao and Kannan, 1976) : Hsiao, D.K., and Kannan, K. 'The Architecture of a Database Computer - Part II: The Design of Structure Memory and its Related Processors'. OSU-CISRC-TR-76-e, Ohio State University, December 1976.

(Huff and Madnick, 1978) : Huff, S.L., and Madnick, S.E. 'An Approach to Constructing Functional Requirement Statements for System Architectural Design', M.I.T. Sloan School of Management, C.I.S.R. Internal Report No. P010-7806-06, (June 1978).

(Johnson, 1975) : Johnson, C. 'IBM 3850 - Mass Storage System'. AFIPS Conference Proceedings, 44, (1975), 509-514.

(Lam and Madnick, 1979) : Lam, C.Y., and Madnick, S.E., 'Properties of Storage Hierarchy Systems with Multiple Page Sizes and Data Redundancy', MIT Sloan School Working Paper No. 1047-79 (also as CISR Working Paper No. 42), 1979.

(Lang et. al., 1977) : Lang, T., Nahouraii, E. Kasuga, K. and Fernadez, E.B. 'An Architectural Extension for a Large Database System Incorporating a Processor for Disk Search'. VLDB, 1977, 204-210.

(Langdon, 78) : Langdon G.G. Jr. 'A Note on Associative Processing for Data Management'. TODS, 3, 2 (June 1978), 148-158.

(Lin et. al., 1976) : Lin, S.C., Smith, D.C.P., and Smith, J.M. 'The Design of a Rotating Associative Memory for Relational Database Applications'. TODS, 1, 1 (March 1976), 53-75.

(Madnick, 1973) : Madnick, S.E. 'Storage Hierarchy Systems'. MIT Project MAC Report No. TR-105, 1973.

(Marill and Stern, 1975) : Marill, T., and Stern, D. 'The Datacomputer - a Network Data Utility'. AFIPS Conference Proceedings, 44 (975), 389-395.

(Ozkarahan et. al., 1975) : Ozkarahan, E.A., Schuster, S.A., and Smith, K.C. 'RAP - Associative Processor for Data Base Management'. AFIPS Conference Proceedings, 44, (1975), 379-388.

(Ozkarahan et. al., 1977) : Ozkarahan, E.A., Schuster, S.A., and Sevcik, K.C., 'Performance Evaluation of a Relational Associative Processor', TODS, 2, 2 (June, 1977), 175-195.

(Schuster et. al., 1976) : Schuster, S.A., Ozkarahan, E.A., and Smith, K.C., 'A Virtual Memory System for a Relational Associative Processor', NCC, 1976, 855-862.

(Senko, 1976) : Senko, M.E. 'DIAM II: The Binary Infological Level and its Database Language - FORAL'. Proceedings of the ACM Conference on Data. March 1976, 1-21.

(Su, 1977) : Su, S.Y.W. 'Associative Programming in CASSM and its Applications'. VLDB, 1977, 213-228.

(Su and Lipovski, 1975) : Su, S.Y.W., and Lipovski, G.J. 'CASSM: A Cellular System for Very Large Databases'. VLDB, September 1975, 456-472.

(Toh et. al., 1977) : Toh, T., Kawazu, S., and Suzuki, K. 'Multi-Level Structures of the DBTG Data Model for an Achievement of the Physical Independence'. VLDB, 1977, 403-414.

(Yeh et. al., 1977) : Yeh, R.T., and Baker, J.W. 'Toward a Design Methodology for DBMS: A Software Engineering Approach'. VLDB, 1977, 16-27.